The Lit Sphere: A Model for Capturing NPR Shading from Art

Peter-Pike J. Sloan, Microsoft Research
William Martin, University of Utah
Amy Gooch, University of Utah
Bruce Gooch, University of Utah

Abstract

While traditional graphics techniques provide for the realistic display of three-dimensional objects, these methods often lack the flexibility to emulate expressive effects found in the works of artists such as Michelangelo and Cezanne. We introduce a technique for capturing custom artistic shading models from sampled art work. Our goal is to allow users to easily generate shading models which give the impression of light, depth, and material properties as accomplished by artists. Our system provides real-time feedback to immediately illustrate aesthetic choices in shading model design, and to assist the user in the exploration of novel viewpoints. We describe rendering algorithms which are easily incorporated into existing shaders, making non-photorealistic rendering of materials such as skin, metal, or even painted objects fast and simple. The flexibility of these methods for generating shading models enables users to portray a large range of materials as well as to capture the look and feel of a work of art. (Color images can be found at http://www.cs.utah.edu/npr/papers/LitSphere_HTML.)

Key words: non-photorealistic rendering, interaction, shading, environment maps, lighting models, painting, paint programs.

1 Introduction

To achieve the range of effects in use today, modelers and animators often engage artists to create custom textures. Currently we lack tools which would allow the non-artist to emulate the nonlinear shading found in works of art. The technical goal of this paper is to create methods for capturing and interactively editing shading models. The artistic goal is to apply to geometric objects and scenes the subtle shading effects artists use to infuse works of art with vitality and beauty. This paper presents algorithms for mapping shading from works of art to geometric models, allowing the user to interactively render scenes based on the style of a reference art work.

When an artist draws or paints an object, they often start with a shading study on the sphere [9]. This is shown for a pencil drawing in Figure 1, where an artist uses a reference drawing of a sphere to draw a head with similar shading. Using the sphere in this way ensures that when

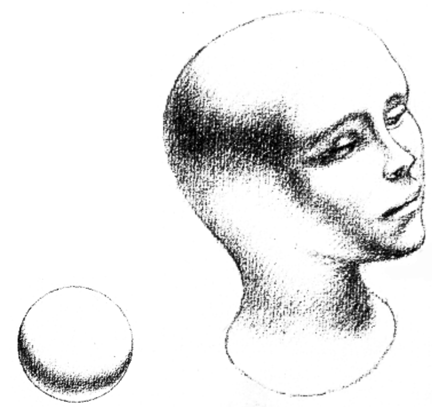


Figure 1: An artist first develops a shading study on a sphere (left) and then adapts this study to a complex object (right). This process allows the artist to first concentrate on color/shading alone, and then to apply the same shading to a complex object in a way that guarantees some global consistency in the final image. Image courtesy of McGraw-Hill Book Company [11].

the complex object is rendered it will have global consistency in its shading. This paradigm of using a reference sphere for shading can be applied in computer graphics. Given an image of a shaded sphere, we can infer how the sphere normals correspond to colors, and shade a model based on local surface orientation.

While transferring a shading model from an image of a sphere to a complex 3D model is straightforward, it would be much more useful to take an image of a complex object and transfer its shading to another complex object. In this paper we use the shading study on a sphere as an intermediate step to help the user "capture" a shading model from an image of a complex object. This essentially inverts the traditional process illustrated in Figure 1.

In Section 2 we review related work and introduce the "lit sphere" model. Section 3 discusses implementation methods for applying the model to non-photorealistic rendering, and describes the user interface. We present results and some applications for our approach in Section 4. We suggest areas for further investigation and conclude the paper in Sections 5 and 6.

2 Background

Our shading algorithms belong to a family of computer graphics techniques collectively referred to as nonphotorealistic rendering (NPR). An underlying assumption in NPR is that artistic techniques developed by human artists have intrinsic merit based on the evolutionary nature of art. For this reason techniques are usually borrowed from artists rather than reinvented from first principles.

Representation of material properties has a long history in the graphics community. Schlick [14] provides a good survey of this literature. Our paper takes a different approach, focusing on reproducing shading algorithms used by an artist in their work. In NPR, only the areas of cartoon shading [10] and technical illustration [5] have incorporated color shading techniques. There have been several researchers outside of the NPR community who have tried to capture lighting effects from one medium and apply them to another scenario. Work by Miller and Hoffman [12], Debevec et al. [3, 4], and Sato et al. [13] captured lighting effects in order to plausibly embed computer graphics objects in photographs or video or to create new scenes under the same environmental conditions. Reverse engineering of BRDFs was applied to the domain of photorealism by Yu et al. [16, 17]. In this paper, we attempt to solve a similar inverse problem to reproduce artistic shading by sampling art work. One notable departure from prior work is that no attempt is made to separate lighting information from texture. This is due to the incomplete information available in a single artwork sample. Our techniques are grounded in the ideas for environment maps presented by Blinn [2] and Greene [7], with slight variations on the mapping to the target surface. The lit sphere approach is similar in spirit to results obtained by Heidrich and Seidel [8].

To understand the proposed method for extracting artistic shading models, it is useful to review techniques utilized in the rendering community for capturing reflectance. Suppose we have a small patch of a given material within an environment, and we wish to measure how much light leaves the surface with respect to a set of outgoing directions. We can accomplish this by keeping the environment and measuring device fixed and rotating the patch about its center. This procedure allows us to determine a correspondence between the direction of the surface normal and the amount of light which reaches the device.

A sphere provides coverage of the complete set of unit normals. Assuming the sphere is small with respect to the scale of the scene, the light arriving at all nearby points

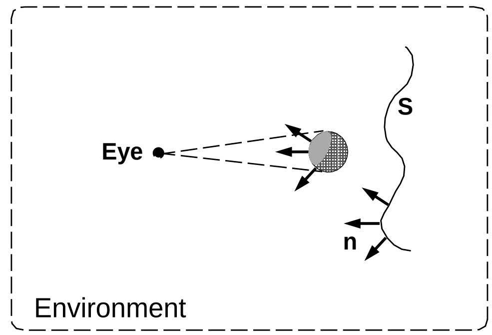


Figure 2: A sphere embedded in an environment acts as a surrogate for the desired surface S, providing a simple correspondence between lighting and the normal $\mathbf{n}$.

on the sphere will be roughly the same, allowing us again to relate surface normal to reflected light for a particular material. Thus, approximate measurements can be obtained by photographing a sphere made of the given material embedded in the identical scene whose radius is small compared to the scale of the scene (see Figure 2).

Finally, suppose we have an object of the same material as the sphere, and similar scale. Under these conditions the reflectance of the surface and the character of the incoming light will not vary much from point to point on the surface. If this is true, then we are justified in replacing the sphere with the desired surface and lighting it according to surface normal using the data from the photographed sphere. In essence, the sphere serves as a surrogate for more complex objects in order to simplify the characterization of reflected light and assure sufficient coverage of normals. We refer to this “paint by normals” method of shading as the *lit sphere* model. Because our model is informed only by light which leaves a surface in the direction of the eye, it will fail to distinguish between variations due to lighting and those due to texture. Therefore, our method will allow the user to render scenes from novel viewpoints, but will exhibit artifacts under animation as the texture swims across the surface to follow the eye. We shall explore the ramifications further in Section 5.

3 Lit Sphere Shading

For the purpose of artistic rendering, it is seldom crucial that the shading be exact. It is more important that shading convey form, be descriptive of texture, and place objects within a scene in a common context. Therefore we can relax some of the assumptions of the lit sphere model, settling for some inexactness in exchange for greater applicability.

If the art work is a shaded sphere, then we can apply the

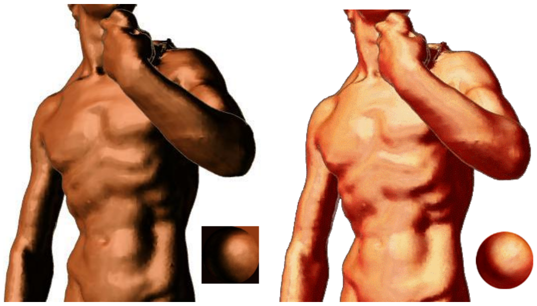


Figure 3: Rendered David model and the corresponding lit spheres. In this instance the spheres were drawn by hand, scanned, and imported into our system.

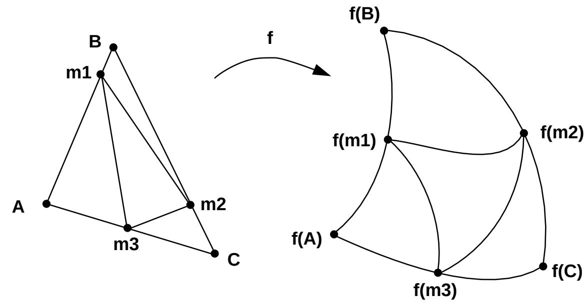


Figure 4: The mapping between planar texture triangles and spherical triangles on the lit sphere.

lit sphere model of the last section directly, using the art work as the "photograph." Figure 3 shows artistically rendered spheres and the resulting lit sphere shading applied to a 3D model. Note that complex shading and indirect lighting effects encoded by the artist transfer naturally to the rendered images.

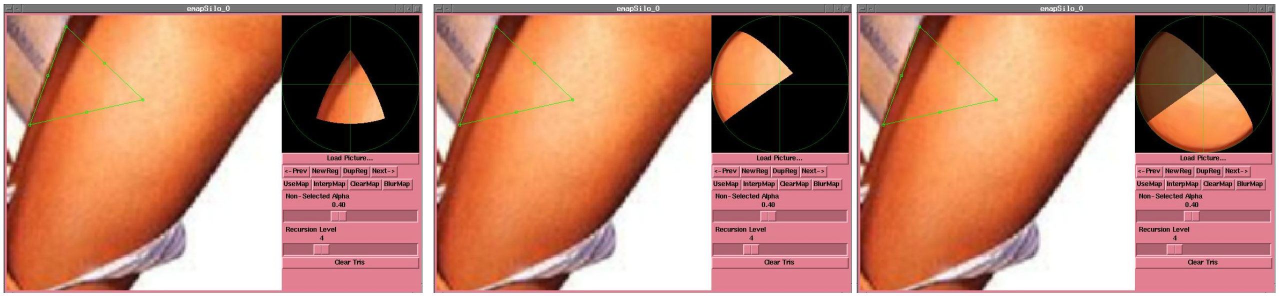
However, it is too restrictive to require that an artistically rendered sphere be produced for each geometric object in the scene. Instead, we derive a method for extracting lit spheres from source art. Suppose that a piece of art work contains surfaces with locally spherical patches. These patches possess an approximately correct distribution of normals. Thus, we can approximate the artistic lighting model by projecting the shaded patch onto a lit sphere. The patch may lack part of the hemispherical normal space or distort the distribution of normals. Thus, our system must provide a method for modifying the mapping from the patch to the lit sphere environment map.

It is easiest to explain our interface by walking through an example, illustrated in Figure 5. A user starts by loading a 2D artistic source image. The user then selects a triangular region in the source image. A corresponding spherical triangle is automatically created on the hemisphere. The user can interactively move and edit both triangles. Moving the planar triangle changes the texture coordinates of the corresponding spherical triangle. Moving the spherical triangle changes the set of normals which correspond to the illumination in the texture. The user can proceed to create triangles and stretch them over the hemisphere until the entire set of normals has been covered and the desired artistic effect is achieved. The effects of all edits can be observed in real time, providing useful feedback to guide the user's choices. Any gaps in the resulting lit sphere are patched using the splat-push-pull method of Gortler et al. [6].

The user should have some control over the parameterization of the image triangles so that the texture can be manipulated to achieve a correct distribution of normals. As illustrated in Figure 4, we provide a simple approximate mechanism that works well in practice. The edge of each planar texture triangle has a "midpoint" which the user can slide along that edge. The midpoints along with the vertices of the planar triangle imply a 4-1 subdivision of the parameterized triangle. This is mapped to a uniform 4-1 subdivision of the spherical triangle, as shown in Figure 4. The result is a piecewise smooth distortion of the texture with $\mathbf{C}^0$ continuity along the boundaries. Any unpleasant artifacts due to the discontinuity along the boundaries can be diminished by applying Gaussian blur to the resulting lit sphere map. The lit sphere maps are stored as images like those seen in Figure 6. The central pixel corresponds to the normal which points toward the eye, and the peripheral edges to silhouettes. Our method employs the SGI environment mapping hardware to achieve real-time display, thereby facilitating user-directed interactive lighting of scenes.

Each object in a scene can be assigned a unique lit sphere map (see Figures 6 and 7). We provide a complete facility for moving between objects, adding, modifying, and deleting both spherical and planar triangles, and for loading image files. For a given object, it is not required that the user choose adjacent triangles or even find the correct distribution of normals in the image. Some interesting effects are possible by violating the assumption that the pieces are drawn from the same locally spherical patch. In the example illustrated in Figure 5 we only needed to choose two triangular regions in the source image, instance those 2D triangles to create four spherical triangles, and then place the spherical triangles on the hemisphere to create a lit sphere.

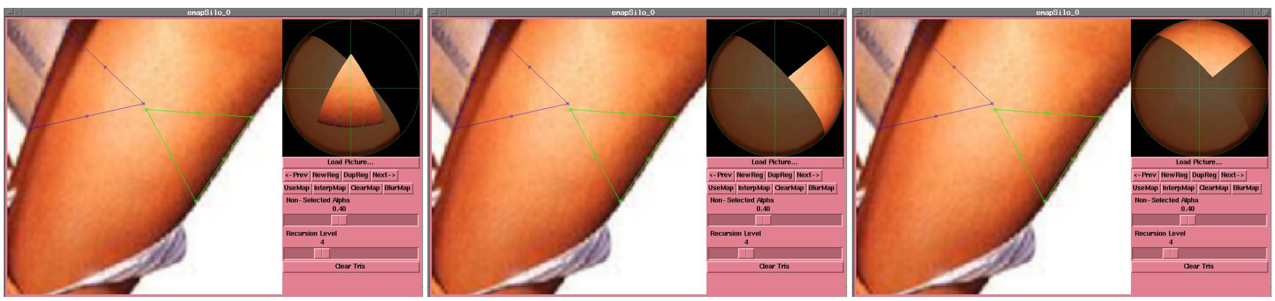
An important issue is that the geometry to which the shading model is targeted may reflect light or cast shadows onto itself. In addition, surfaces in the scene may be close, whereas the surfaces in the source art from which



(a) Load a 2D source image [1] and select a triangular region from the 2D source image (left window). A spherical triangle is automatically created on the hemisphere (upper right window).

(b) Move the corresponding spherical triangle into place on the hemisphere.

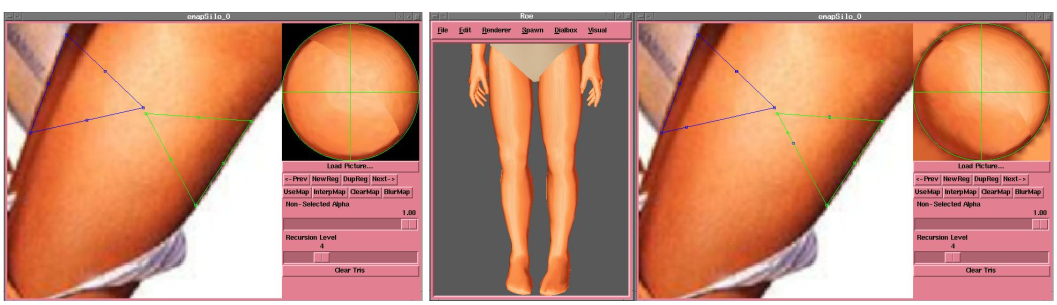
(c) Our interface allows the user to create additional spherical triangles, mapped from the same 2D triangle created previously. Now the user moves the newly created spherical triangle into place.



(d) Our interface also allows users to select a new region in the 2D source image, which also automatically creates a spherical triangle on the hemisphere.

(e) Now the user moves the corresponding spherical triangle into place.

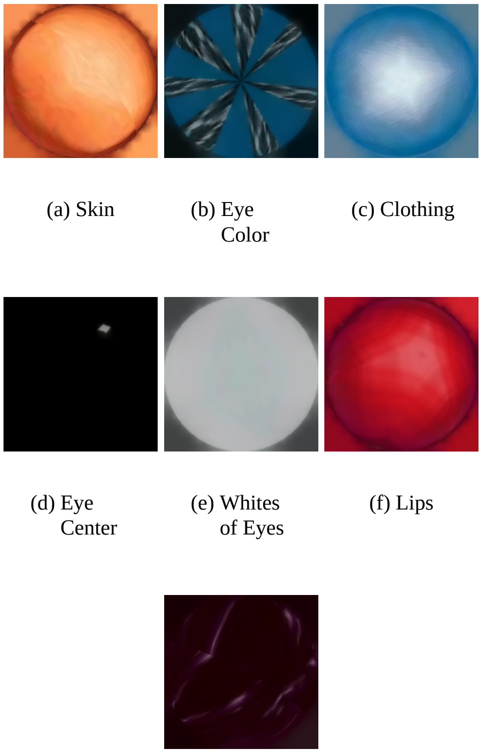
(f) Again the user instances the previously created 2D triangle and moves the newly created corresponding spherical triangle into place on the hemisphere.



(g) We have covered the entire hemisphere, but need to minimize the discontinuity between regions.

(h) The mapping between planar and spherical triangles can be changed by sliding the midpoints on the triangles in the 2D source image. These sliders can be used to exagerate shading at the boundary of the triangles, or to help create some continuity between spherical triangles. We can then apply the lit sphere to the appropriate parts of the model.

Figure 5: A sequence of operations for creating a particular lit sphere.

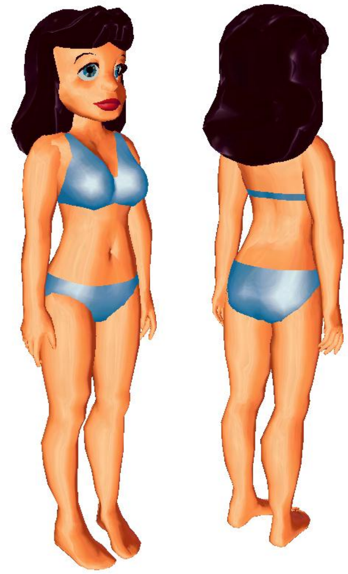


(a) Skin (b) Eye Color (c) Clothing

(d) Eye Center (e) Whites of Eyes (f) Lips

(g) Hair

Figure 6: Seven lit spheres used for shading the model in Figure 7.



(a) (b)

Figure 7: Final shaded girl viewed from two novel viewpoints.

the illumination models were drawn may be distant, or vice versa. Thus, indirect lighting effects may be incompatible. Furthermore if several art sources are utilized, the artistic illumination models may be inconsistent.

We leave it to the user to sort out which assumptions can be relaxed for their application, and provide real-time feedback to guide their choices. We have found that sensible violations do not degrade the resulting output image, allowing the user to exercise a large degree of creative freedom in the selection of source art work.

4 Results

One of the key results of our approach is that non-artists can easily produce images such as those presented in this paper. We tested our interface on three novice users. We first demonstrated the software on the skin images and doll model, shown in Figure 5. We showed how one can select two triangles, duplicate them, and produce a hemisphere which darkens at the silhouettes and has soft highlights in the center. The users were also directed how to use the sliders on the spherical triangles to minimize the discontinuities between triangles. The tutorial for placing the skin on the doll model took 5 minutes from start to finish (see Figure 5). We observed that users are adept at determining the approximate light position in the 2D image and had no problem carrying out the creation of the lit sphere. In ten minutes, one user produced the image in Figure 8 by simply picking an approximately spherical region in the 2D image and forming a circle of triangles which mapped directly onto the hemisphere.

The ability to capture 3D shading models from 2D sources gives great flexibility to the user. With our system it is straightforward to capture shading from art work and reproject it onto geometry. For example, consider the Cezanne painting in Figure 9. We have created a coarse model of the scene, and captured a lit sphere environment map for each scene element from the original art work. Using this mapping, we can rotate the scene and achieve a plausible artistic rendering from the new viewpoint (Figure 10). Also, observe that from the original viewpoint the indirect lighting on the metal cup due to the adjacent lemon is clearly visible.

The flexibility of the input used for creating shaders allows users to create their own palette. For example, consider Figure 11. We start with a palette of black and white. Following the work on non-photorealistic metal in technical illustration by Gooch et al. [5], it is straightforward to achieve the impression of brushed metal using a lit sphere map. An important characteristic of such metals is asymmetry of reflectance under rotation, termed anisotropy. In this example, we added two spherical triangles containing noise and an asymmetrical highlight to obtain the results in Figure 12.

5 Future Work

In this work, we have only considered rendering static scenes with illumination derived from a lit sphere. If we transform the scene as part of an animation, for example, some problems present themselves.

The lit sphere model discussed in Section 2 is based on the assumption that source materials are homogeneous, while artists often encode local surface features in their work. As a result, local material effects may be encoded in the lit sphere model as variation in illumination. If the animation is created by reprojecting the lit sphere at every frame according to the new viewpoint, then local texture features will appear to follow the eye. There may be cases where this effect is unobtrusive.

As an example of such a situation, consider capturing idealized skin such as that seen in Olivia De Berardinis' art work, the "cheesecake portraits" [1]. The shading darkens at the silhouettes and has a soft highlight in the middle. For a 3D model such as the doll in Figure 7, the lit sphere model is convincing when reprojected. As the model moves, the shading will always darken at the silhouettes with a soft highlight in the center.

However, if local texture features are very prominent, the surface will appear to "swim." As an example, we have applied the shading from the frescos in the Sistine Chapel by Michelangelo to a 3D model of Michelangelo's David (Figure 8). There is a lot of texture information in the image of the fresco. When we apply the lit sphere and rotate the model, David appears metallic or satiny due to the high frequency noise. We'd prefer to make the texture stick to the object and allow the lighting to follow the eye, similar to lit sphere applied to the doll.

For completeness, we will briefly describe an approach to achieve approximate separation of texture and lighting. The first step is to extract lighting from the hemisphere. In many cases, we have found that a band-pass filter can be used successfully to extract variation in shading due to lighting. Another approach is to apply the work of Walter et al. [15] to fit virtual Phong lobes to the surface of the lit hemisphere, and subtract out their contributions. The result of either method will be two hemispheres, the first containing the lighting information, the second being the texture information that remains when the lighting

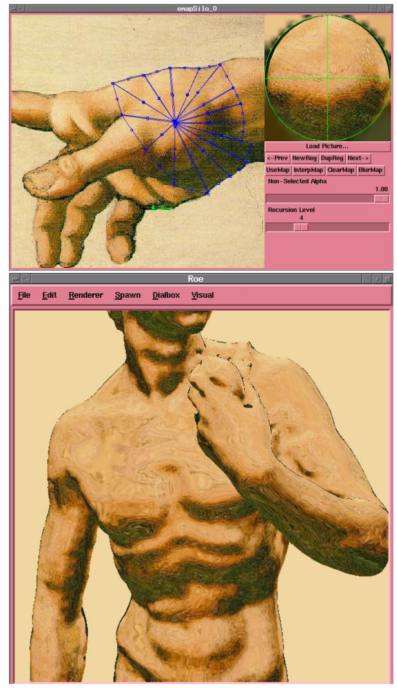


Figure 8: Rendering according to a lit sphere captured from a portion of Michelangelo's Sistine Chapel ceiling. The shading model was created by sampling a ring of triangles in the source image.

is excised. Because lighting often obscures local texture features, the second sphere cannot be used directly as a texture map. Furthermore, we have texture information only for the front side of the sphere. This leaves half the surface bare. The most generally applicable technique seems to be synthesizing the texture over the sphere using the information in the lit sphere minus lighting. Synthesis directly on the sphere would also serve to avoid the distortions which occur when a planar image is mapped onto the sphere.

Rather than draw attention away from the simple idea which is at the core of this paper, we relegate further exploration of these methods to future work. It has been our experience that the use of lit spheres in rendering from novel viewpoints works without qualification. The quality of animations will depend on the degree to which the captured spheres meet the requirements of the lit sphere model (for example, that the represented materials are homogeneous).

Future research could connect the ability to create lit spheres with a 3D paint program. This would provide a quick underlying surface texture, over which the user could then add features that cannot be uniformly applied

Figure 9: Still life by Cezanne.

over the model, such as detail brush strokes or features such as blush or freckles on a face. In addition, automatic methods for determining shading models from any artistic image may be possible.

6 Conclusion

Our approach allows material properties found in 2D art work to be retargeted to 3D geometric models. This work incorporates established principles from the art, computer vision, and rendering communities into a framework for non-photorealistic rendering. Our goal is to leverage art work to obtain effects which may be difficult or impossible to produce with current shading methods. We provide an interface which can be used to tailor the artistic shading model to the aesthetic preferences of a user. Our method allows the user to interactively explore novel viewpoints of 3D models, capturing the light and material properties encoded by artists in 2D imagery.

Figure 10: Model illuminated by lit spheres captured from the Cezanne still life.

Acknowledgments

Thanks to Michael Stark, Peter Shirley, Louise Bell, Grue, Richard Coffey, Susan Ashurst, Matthew Kaplan, James de St. Germain, and the members of the University of Utah Computer Graphics groups for help in the initial stages of the paper. Thanks also to Michael Cohen, Adam Finkelstein, James Mahoney, and Wolfgang Heidrich for comments and critical review. Thanks to Viewpoint Data Labs for the girl model and to the University of Utah Alpha_1 group for the crank shaft model. Thanks to Mark Harden (www.artchive.com) for the scanned images of Michelangelo's "Creation" and for Cezanne's still life. This work was supported in part by DARPA (F33615-96-C-5621) and the NSF Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-89-20219). All opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

[1] Joel Beren and Olivia De Berardinis. *Let Them Eat Cheesecake*. Ozone Productions, Ltd, Roslyn, NY, 1993.

[2] J. F. Blinn and M. E. Newell. Texture and Reflection in Computer Generated Images. *Communications of the ACM*, 19:542–546, 1976.

[3] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. *Proceedings of SIGGRAPH 98*, pages 189–198, July 1998. ISBN 0-89791-999-8. Held in Orlando, Florida.

[4] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. *Proceedings of SIGGRAPH 2000*, pages 145–156, July 2000. ISBN 1-58113-208-5.

[5] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A Non-photorealistic Lighting Model for Automatic Technical Illustration. In *Computer Graphics*, July 1998. ACM Siggraph '98 Conference Proceedings.

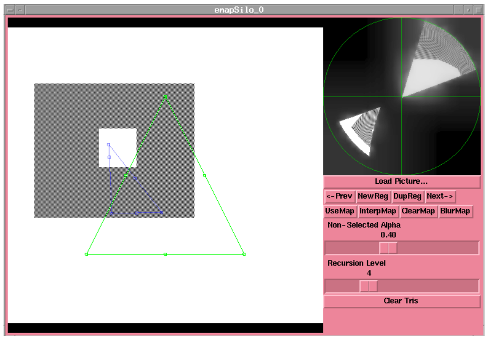


Figure 11: The creation of an NPR metal shader from a simple palette of black and white.

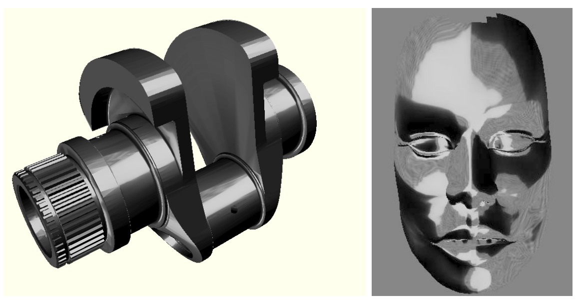


Figure 12: A crank part and a mask illuminated according to the simple noisy lit sphere shown in Figure 11.

[6] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The Lumigraph. In *Computer Graphics Proceedings, Annual Conference Series, 1996*, pages 43–54, 1996.

[7] Ned Greene. Environment Mapping and Other Applications of World Projections. *IEEE Computer Graphics and Applications*, 6(11):21–29, November 1986.

[8] Wolfgang Heidrich and Hans-Peter Seidel. Realistic, hardware-accelerated shading and lighting. *Proceedings of SIGGRAPH 99*, pages 171–178, August 1999. ISBN 0-20148-560-5. Held in Los Angeles, California.

[9] Ted Seth Jacobs. *Light for the Artist*. Watson Guptill Publications, 1988.

[10] Adam Lake, Carl Marshall, Mark Harris, and Marc Blackstein. Stylized rendering techniques for scalable real-time 3d animation. In *Non-Photorealistic Animation and Rendering 2000 (NPAR '00)*, Annecy, France, June 5-7,2000.

[11] Edward Laning. *The Act of Drawing*. McGraw-Hill, New York, 1971.

[12] Gene S. Miller and C. Robert Hoffman. Illumination and reflection maps: Simulated objects in simulated and real environments. *SIGGRAPH '84 Advanced Computer Graphics Animation seminar notes*, July 1984.

[13] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):1–12, January - March 1999.

[14] Christophe Schlick. A Survey of Shading and Reflectance Models. *Computer Graphics Forum*, 13(2):121—131, June 1994.

[15] Bruce Walter, Gun Alppay, Eric P. F. Lafortune, Sebastian Fernandez, and Donald P. Greenberg. Fitting Virtual Lights for Non-Diffuse Walkthroughs. In *SIGGRAPH 97 Conference Proceedings*, pages 45–48, August 1997.

[16] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. *Proceedings of SIGGRAPH 99*, pages 215–224, August 1999. ISBN 0-20148-560-5. Held in Los Angeles, California.

[17] Yizhou Yu and Jitendra Malik. Recovering Photometric Properties of Architectural Scenes from Photographs. *SIGGRAPH 98 Conference Proceedings*, July 1998.